



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/714,018	11/14/2003	Simon Pizzoli	GB920030026US1	1974
23550	7590	03/19/2009		
HOFFMAN WARNICK LLC 75 STATE STREET 14TH FLOOR ALBANY, NY 12207			EXAMINER NGUYEN, PHILLIP H	
			ART UNIT 2191	PAPER NUMBER
			NOTIFICATION DATE 03/19/2009	DELIVERY MODE ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

PTOCommunications@hoffmanwarnick.com



UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

**BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES**

Application Number: 10/714,018
Filing Date: November 14, 2003
Appellant(s): PIZZOLI ET AL.

Meghan Q. Toner
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 12/09/2008 appealing from the Office action mailed 07/22/2008.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

5,475,843	Halviatti et al.	12-1995
6,167,534	Straathof et al.	12-2000

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims and is the Examiner's ultimate position:

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

2. Claims 1-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Halviatti et al. (USPN 5,475,843), in view of Straathof et al. (USPN 6,167,534).

As per claims 1, 10 and 19:

Halviatti discloses a system for language-neutral user interface automation, the system comprises:

a system executing a test, the system executing the test including a processor (see at least **FIG. 1A**; see also col. 5:26-27 "**system 100 which comprises a central processor 101**"); and

a memory (see at least **FIG. 1A**; see also col. 5:26-27 "**system 100 which comprises...a main memory**"), the memory including:

automation script means (see at least col. 8:51 "**Script Engine 330**") for receiving an automation script for automating use of the user interface in a system under test by the system executing the test (see at least col. 8:58-64

"Messages from these events, including system messages 310 and target application messages, are trapped by the ATU 340 and reported to the Message Engine 350 as CBT messages. The Message Engine, in turn, dispatches the message according to handlers specified by the Script Engine 330, operating under the control of script 320"), wherein the system under test include an application (see at least FIG. 1B; see also col. 5:41-49 ***"a computer software system 150 is provided for directing the operation of the computer system 100...includes a window shell 180. One or more application programs, such as application software 170...system 150 includes a user interface (UI) 165 preferably a graphical user interface, for receiving user commands and data"***) and wherein the interface may be in a arbitrary natural language (see at least ***FIG. 1C – The text displayed are in English language***); and

script translation means for intercepting a call from the automation script to a function simulating a user action on the application (see at least ***FIG. 3***; see also col. 9:20-21 ***"each ATU functions to trap events and convert them into "CBT messages"***; see also col. 8:58-60 ***"Messages from these events, including system message 310 and target application messages, are trapped by the ATUs (Application Translation Units) 340"***),

retrieving a translated text string associated with the function call (see at least col. 7:55-65 ***"where a message (&msg) is retrieved by a call to GetMessage, it needed, the retrieved message may be translated by a call***

to TranslateMessage() and then dispatched by a call to DispatchMessage()), and

calling the function simulating the user action with the translated text string (see at least FIG. 3 – “**simulate user access**”; see also at least 7:13-16 “**In a typical environment, especially those typified by a character-based UI, a program from the keyboard by making an explicit call to a function, such as the C function getchar(). The function typically waits until the user presses a key before returning the character code to the program**”; see also col. 33:25-30 “**class methods are provided for simulating user events. For instance, the Click(), DbClick()...**”; see also col. 34:16 “**user interaction simulation, and Resource tracking**”; further see also col. 35:38-44 “**Each GEM can simulate any possible operation that a user would perform on any given element...**”); and

wherein the translation consists of converting **to or from** a first natural language to a second natural language (see at least col. 9:20-24 “**whether application specific or not, each ATU functions to trap events and convert them into “CBT message” - a lingua franca or common language for all events, whether Windows application-specific, occurring within the system**”; see also col. 30:47-55 “**By storing object attributes in a database, the need for “hard coding” textual or positional information within ATMs for within test scripts is eliminated. This is particularly advantageous because any textual or positional assumptions about a UI element often**

change, especially when the application is to be localized (i.e. translated to different languages) for several international versions. In this manner, the need for altering the ATMs or the test scripts every time the text label for a UI element changes is eliminated").

Halviatti does not explicitly teach:

wherein the interception includes accessing a database or file system that is independent from the system under test so that the application's natural run-time execution is protected before, during and after the functional automation executes.

However, Straathof teaches:

wherein the interception includes accessing a database or file system that is independent from the system under test so that the application's natural run-time execution is protected before, during and after the functional automation executes (see at least col. 5:55-67 – col. 6:38-47 ***"the database application determined that the user interface call is a request to logon to the database application...the database application received a user interface call requesting to insert data into the database...received a user interface call requesting to fetch data from the database...An important aspect of the present invention is the Capture Agent. A Capture Agent 262 captures one or both of the Windows and SQL API calls during a user session. The***

Capture Agent not only intercepts the user interface and application calls, the Capture Agent records timing information regarding when the calls were sent. This allows the Capture Agent to generate a script 264 to emulate the user session including the speed in which the user input information and the speed in which the client computer responded locally"

– Note that the database is located on a server and because it located separately on the server, the application's natural run-time execution is protected).

Therefore, it would have been obvious to one having an ordinary skill in the art at the time the invention was made to incorporate the teaching of Straathof into Halviattis' approach to allow intercepting by accessing a database or file system that is independent from the system under test. The modification would have been obvious to one of ordinary skill in the art because it would allow generating a script to emulate the user session including the speed in which the user input information and the speed in which the client computer responded locally.

As per claims 2, 11 and 20:

Halviatti further discloses:

message translation means for supplying translated text for the automation script's run time execution (see at least **FIG. 3, item 340 – "Application Translation Unit(s) 340 comprises Translators"**; see also col. 7:64 **"TranslateMessage()"**); and

selective text locator means coupled to the message translation means for selectively supplying appropriately translated text to the automatically script's run time execution depending on the function call in a case that a same text string is translated differently based on context (see at least col. 7:55-67

```
"while (GetMessage(&msg, NULL,0,0))
```

```
{
```

```
    TranslateMessage(&msg) ;
```

```
    DispatchMessage(&msg) ;
```

```
}
```

```
return msg.wParam;
```

```
}
```

Where a message (&msg) is retrieved by a call to GetMessage() if needed, the retrieved message may be translated by a call to TranslateMessage() and then dispatched by a call to DispatchMessage()"

As per claims 3 and 12:

Halviatti discloses:

wherein the selective text locator means is arranged to selectively supply appropriate text to the automation script's run time execution depending on a resource ID of the function call (see at least col. 37:4-10 ***"an unique id is constructed for this top level menu...the resource database is search for***

the record under this id...the tope level menu string from the menu handle is retrieved...a preferred name is also retrieved...").

As per claims 4 and 13:

Halviatti discloses:

a library (see at least col. 31:21 "**GEM Library**" – a collection of GEMs) including a function having the same signature as the function call and which is arranged to retrieve the translated text string before the function call (see at least col. 31:60 – col. 32:1-60 "**a GEM encapsulates the behavior of irreducible user interface elements such as push buttons, checkboxes, listboxes, menu items...when GEM is instantiated, it takes two parameters...the GEM can be instructed to run a self test method by simple comparing its expected attributes against its actual attributes (retrieved from the actual element on the screen which the GEM represents)**"; and

one of a file ("**a GEM**" col. 31:43) referencing the library (see at least col. 31:42-44 "**when a GEM instantiated, it takes two parameters: a reference to its parent and...**" - a GEM is also considered as a file or a program, it references to its parent, a library), the automation script being arranged to reference the file and the library (see at least col. 41:15-20 "**Upon invocation of a test script, the Test Runtime Library is loaded, the resource database is opened and initialized the GEM library classes are loaded...**" – in other words, when the a test script is being tested, the GEM library classes gets called), and the library

including the retrieval function (see at least col. 31:51-55 "***self test method is comparing its expected attributes against its actual attributes (retrieved from the actual element on the screen which the GEM represents)***") and the function call (see at least col. 31:43-44 "***a GEM ... a reference to its parent***" - a call to its parent) , the library being arranged to be called by the automation script (see at least col. 41:15-20 "***Upon invocation of a test script, the Test Runtime Library is loaded, the resource database is opened and initialized the GEM library classes are loaded...***" - in other words, when the a test script is being tested, the GEM library classes get called).

As per claims 5 and 14:

Halviatti discloses:

wherein the file referencing the library comprises an include file (see col. 49 "***Appendix E: Window Proxy***").

As per claims 6 and 15:

Halviatti discloses:

wherein the library including the retrieved function and the function call has the same name as a library containing the function called by the automation script (see col. 31:21 "***GEM Library***" – *GEM Library including retrieved function and function call disclosed above and is called by automation script upon invocation of a test script*).

As per claims 7 and 16:

Halviatti discloses:

wherein the automation script comprises a script (see at least col. 10:36

"script").

As per claims 8 and 17:

Halviatti discloses:

wherein the automation script is in the English language and the

application is arranged to use a non-English language (see at least col. 9:20-22

"a lingua franca or common language for all events").

As per claims 9 and 18:

Halviatti discloses:

wherein the user interface comprises a graphical user interface (see at

least ***FIG. 1***, and text, which further expand their features, col. 5:51).

(10) Response to Argument

Appellant argues claims 1, 2, 10, 11, 19, and 20

1) Halviatti fails to teach "wherein the translation consists of converting to or from a first natural language to a second natural language"

2) Halviatti relies on the specific interception of the running application to achieve its goals, therefore, not protecting the application's natural run-time execution. The interception in Straathof does not protect the application's natural run-time execution because the hook used in Straathof functions between the application and the system, and therefore inherently changes the natural behavior of the application. Thus, Halviatti and Straathof fail teach "script translation means for intercepting a call from the automation script to a function simulating a user action on the application, wherein the interception includes accessing a database or file system that is independent from the system under test so that the application's natural run-time execution is protected before, during and after the functional automation executes."

3) In Halviatti, the general call to translate a message in a Windows environment is not equivalent to selecting an accurate text translation from a plurality of available translations. Therefore, Halviatti fails to teach "selective text locator means coupled to the message translation means for selectively supplying appropriately translated text to the automation script's run time execution depending on the function call in a case that a same text string is translated differently based on context."

4) Claim 7 is not indefinite because the term JAVA in claim 7 is used to describe the language in which the script is written.

5) Claims 19 and 20 are not being directed to non-statutory subject matter.

Examiner respectfully disagrees:

1) The recited limitation in independent claims does not limit to translate from a first natural language to a second natural language. Instead, it only requires translating to a second natural language regardless of the first language is natural language or not. Based on the claim language, it is reasonable to interpret this limitation as "wherein the translation consists of converting to a second natural language." Halviatti teaches this limitation in col. 9:20-24 ***"whether application specific or not, each ATU functions to trap events and convert them into "CBT message" - a lingua franca or common language for all events, whether Windows application-specific, occurring within the system."*** According to Halviatti, the user events are translated to natural language such as a lingua franca or common language. Halviatti goes on to teach in col. 30:47-55 ***"By storing object attributes in a database, the need for "hard coding" textual or positional information within ATMs for within test scripts is eliminated. This is particularly advantageous because any textual or positional assumptions about a UI element often change, especially when the application is to be localized (i.e. translated to different languages) for several international versions. In this manner, the need for altering the ATMs or the test scripts every time the text label***

for a UI element changes is eliminated". Thus, it is clear that Halviatti's approach concerns with the natural languages translation.

2) Although the appellant explains that the level of interception in Halviatti is intrusive and has the disadvantage of creating a high probability of composing the natural execution of the application but this is not based on fact. An ordinary skill in the art does not know why or how the interception in Halviatti interferes with the natural execution of the application. Moreover, as recited in the independent claims that intercepting by accessing a database or file system that is independent from the system under test could avoid the interference of the natural execution of the application. The interception in Straathof includes accessing a server database that is independent from the system (see at least col. 5:55-67 – col. 6:38-47). Thus, Straathof's system protects the natural execution of the application and the combination of Halviatti with Straathof teaches the claimed invention.

3) The GetMessage() in Halviatti retrieves the messages not the TranslateMessage(). The TranslateMessage() provides translated messages if needed. In other words, GetMessage() retrieves messages and then translated by the TranslateMessage() if needed. The GetMessage() coupled to the TranslateMessage() to return an appropriate translated text. Since, GetMessage() returns an appropriate translated text, it is the same as selective text locator as recited in the claim. Furthermore, the appellant indicates that a general call to translate a message in a windows environment is not equivalent to selecting an accurate text translation from a plurality of available translation—this is not based on fact.

- 4) The rejection to claim 7 is hereby withdrawn.
- 5) The rejection to claims 19 and 20 is hereby withdrawn.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Phillip H. Nguyen, Examiner

Conferees:

/Wei Y. Zhen/

Wei Zhen, Supervisor Patent Examiner, Art Unit 2191

/Lewis A. Bullock, Jr./

Lewis Bullock, Supervisor Patent Examiner, Art Unit 2193